# Approximation for Maximum Surjective Constraint Satisfaction Problems

Walter Bach
CNRS/LIX, École Polytechnique, France
bach@lix.polytechnique.fr

Hang Zhou
Département d'Informatique, École Normale Supérieure, France
hang.zhou@ens.fr

September 21, 2011

**Abstract**

Maximum surjective constraint satisfaction problems (Max-Sur-CSPs) are computational problems where we are given a set of variables denoting values from a finite domain $B$ and a set of constraints on the variables. A solution to such a problem is a *surjective* mapping from the set of variables to $B$ such that the number of satisfied constraints is *maximized*. We study the approximation performance that can be achieved by algorithms for these problems, mainly by investigating their relation with Max-CSPs (which are the corresponding problems without the surjectivity requirement). Our work gives a complexity dichotomy for Max-Sur-CSP($\mathcal{B}$) between PTAS and APX-complete, under the assumption that there is a complexity dichotomy for Max-CSP($\mathcal{B}$) between PO and APX-complete, which has already been proved on the Boolean domain and 3-element domains.

## 1 Introduction

The constraint satisfaction problem (CSP) is an important computational problem in NP where the task is to decide whether there exists a homomor-

phism from a given input relational structure $\mathcal{A}$ to a fixed template relational structure $\mathcal{B}$, denoted by CSP($\mathcal{B}$). This problem has many applications in various fields of computer science, such as artificial intelligence and database theory. It is conjectured that CSP($\mathcal{B}$) for any finite structure $\mathcal{B}$ is either in P or NP-complete [7]. This conjecture has been proved for the Boolean domain [16] and 3-element domains [3].

The optimization version of CSP($\mathcal{B}$), denoted by MAX-CSP($\mathcal{B}$), consists in finding a function from $\mathcal{A}$ to $\mathcal{B}$ satisfying the maximum number of constraints in $\mathcal{A}$ (see, e.g., [5]). It is conjectured that for any finite $\mathcal{B}$, MAX-CSP($\mathcal{B}$) is either in PO or APX-complete [5]. This conjecture has been proved for the Boolean domain [5] and 3-element domains [9].

The surjective version of CSP($\mathcal{B}$), denoted by SUR-CSP($\mathcal{B}$), consists in finding a homomorphism using all available values in $\mathcal{B}$ (see, e.g., [2]). Some problems of the form SUR-CSP($\mathcal{B}$) have been proved to be NP-hard, such as SUR-CSP($C_4^{\mathrm{ref}}$) [13]; some have unknown computational complexity, such as SUR-CSP($C_6$) [6] and the No-Rainbow-Coloring problem [1].

In this article we are interested in the combination of the above two branches of research, that is, the *maximum surjective constraint satisfaction problem* for $\mathcal{B}$, denoted by MAX-SUR-CSP($\mathcal{B}$), which consists in finding a surjective function from $\mathcal{A}$ to $\mathcal{B}$ satisfying the maximum number of constraints in $\mathcal{A}$. There are many natural computational problems of the form MAX-SUR-CSP($\mathcal{B}$). For example, given a graph with red and blue edges, we might ask for a non-trivial partition of the vertices into two sets such that the sum of the number of red edges between the two sets and the number of blue edges inside the first set is minimized. This corresponds to the MAX-SUR-CSP($\mathcal{B}$) where $\mathcal{B}$ is the relational structure on the Boolean domain with two binary relations: equality and disjunction. We will see this problem later, which is known the *Minimum-Asymmetric-Cut problem*.

It is easy to see (see Proposition 7) that all MAX-SUR-CSPs are in the complexity class APX. Our main result is a complexity dichotomy for MAX-SUR-CSP($\mathcal{B}$) on the Boolean domain and 3-element domains: every such problem is either APX-complete or has a *PTAS (Polynomial-Time Approximation Scheme)*. Interestingly, unlike MAX-CSP($\mathcal{B}$), there are finite structures $\mathcal{B}$ such that MAX-SUR-CSP($\mathcal{B}$) is NP-hard but has a PTAS.

The article is organized as follows. In Section 2, we formally introduce MAX-SUR-CSPs. In Section 3, we show that MAX-SUR-CSP($\mathcal{B}$) is in APX for any finite $\mathcal{B}$, by providing a constant-ratio approximation algorithm. In Section 4 and 5, we compare the approximation ratio of MAX-SUR-CSP($\mathcal{B}$)

with that of Max-CSP($\mathcal{B}$). This comparison leads to the inapproximability or approximability for Max-Sur-CSP($\mathcal{B}$), which depends on the desired approximation ratio. In Section 6, we conclude that if there is a complexity dichotomy for Max-CSP($\mathcal{B}$) between PO and APX-complete, there is also a complexity dichotomy for Max-Sur-CSP($\mathcal{B}$) between PTAS and APX-complete. An immediate consequence of this observation is the complexity dichotomy for Max-Sur-CSP($\mathcal{B}$) on the Boolean domain and 3-element domains. In Section 7, we discuss for which structures $\mathcal{B}$ the Max-Sur-CSP($\mathcal{B}$) problem is in PO and pose several open problems.

# 2 Preliminaries

Let $\sigma$ be some finite relational signature, which consists of $m$ relations $R_1, ..., R_m$ of arity $k_1, ..., k_m$ respectively. When $m = 1$, we write $R$ for $R_1$ and $k$ for $k_1$. Consider only finite $\sigma$-structures $\mathcal{A} = (A, R_1^{\mathcal{A}}, \ldots, R_m^{\mathcal{A}})$ and $\mathcal{B} = (B, R_1^{\mathcal{B}}, \ldots, R_m^{\mathcal{B}})$, where $A$ and $B$ are underlying domains, and $R_i^{\mathcal{A}}$ and $R_i^{\mathcal{B}}$ are relations on $A$ and $B$ respectively, $i \in \{1, \ldots, m\}$. A *homomorphism* from $\mathcal{A}$ to $\mathcal{B}$ is a function $h : A \to B$ such that, $(a_1, \ldots, a_{k_i}) \in R_i^{\mathcal{A}}$ implies $(h(a_1), \ldots, h(a_{k_i})) \in R_i^{\mathcal{B}}$, for every $i \in \{1, \ldots, m\}$. Define $|\mathcal{A}|$ to be $|A| + |R_1^{\mathcal{A}}| + \cdots + |R_m^{\mathcal{A}}|$.

The *constraint satisfaction problem* CSP($\mathcal{B}$) takes as input some finite $\mathcal{A}$ and asks whether there is a homomorphism $h$ from $\mathcal{A}$ to the fixed template $\mathcal{B}$. The *surjective constraint satisfaction problem* Sur-CSP($\mathcal{B}$) is defined similarly, only we insist that the homomorphism $h$ be surjective. The *maximum constraint satisfaction problem* Max-CSP($\mathcal{B}$) asks for a function $h$ from $A$ to $B$, such that the number of constraints in $\mathcal{A}$ preserved by $h$ is maximized. The *maximum surjective constraint satisfaction problem* Max-Sur-CSP($\mathcal{B}$) is defined similarly, only we insist that the function $h$ be surjective. Without loss of generality, we assume that $|A| \geq |B|$ in Max-Sur-CSP($\mathcal{B}$), otherwise there is no surjective function from $A$ to $B$.

**Example 1.** Define the relational structure $\mathcal{B}$ on the Boolean domain with two binary relations: equality and disjunction.

Sur-CSP($\mathcal{B}$) is in P, since it has a simple polynomial-time algorithm as follows. Define a connected component to be the maximal set of variables connected together by the equality relation. If there exists some connected component without any disjunction relation between its variables, then we

assign 0 to all its variables and 1 to all other variables. Otherwise there is no surjective assignment.

We call MAX-SUR-CSP($\mathcal{B}$) the *Minimum-Asymmetric-Cut problem*. In fact, if there is only the equality relation, this problem reduces to the *Minimum-Cut problem*. The disjunction relation makes this cut asymmetric.

**Example 2.** Define the relational structure $\mathcal{B}$ on the domain $B = \{0, \ldots, 5\}$ with the binary relation

$$R^{\mathcal{B}} = \{(x, y) \,:\, x - y \bmod 6 = \pm 1\}.$$

This structure is traditionally denoted by $C_6$ in graph theory. The complexity of the *list homomorphism problem* on this structure has been studied in [6]. The complexity of SUR-CSP($C_6$) is still open [2].

**Example 3.** Define the relational structure $\mathcal{B}$ on the domain $B = \{0, 1, 2, 3\}$ with the binary relation

$$R^{\mathcal{B}} = \{(x, y) \,:\, x - y \bmod 4 = 1 \text{ or } 0\}.$$

This structure is denoted by $C_4^{\mathrm{ref}}$.

SUR-CSP($C_4^{\mathrm{ref}}$) has been proved to be NP-complete [13].

**Example 4.** Define the relational structure $\mathcal{B}$ on the domain $B = \{0, 1, 2\}$ with the ternary relation

$$R^{\mathcal{B}} = \{0, 1, 2\}^3 \backslash \{(x, y, z) \,:\, x, y, z \text{ distinct}\}.$$

SUR-CSP($\mathcal{B}$), also known as the *No-Rainbow-Coloring problem*, has open complexity [1]. We call MAX-SUR-CSP($\mathcal{B}$) the *Minimum-Rainbow-Coloring problem.*

An optimization problem is said to be an *NP optimization problem* [5] if instances and solutions can be recognized in polynomial time; solutions are polynomial-bounded in the input size; and the objective function can be computed in polynomial time. In this article, we consider only maximization problems of this form, i.e., *NP maximization problems.*

**Definition 1.** A solution to an instance $\mathcal{I}$ of an NP maximization problem $\Pi$ is *$r$-approximate* if it has value $Val$ satisfying $Val/Opt \geq r$, where $Opt$ is

the maximal value for a solution of $\mathcal{I}$.[1] A *polynomial-time $r$-approximation algorithm* for an NP maximization problem $\Pi$ is an algorithm which, given an instance $\mathcal{I}$, computes an $r$-approximate solution in time polynomial in $|\mathcal{I}|$. We say that $r$ is a *polynomial-time approximation ratio* of an NP maximization problem $\Pi$ if there exists a polynomial-time $r$-approximation algorithm for $\Pi$.

The following definitions can all be found in [5], only with a different convention of the approximation ratio.

**Definition 2.** An NP maximization problem is in the class *PO* if it has an algorithm which computes the optimal solution in polynomial time; and is in the class *APX* if it has a polynomial-time $r$-approximation algorithm, where $r$ is some constant real number in $(0, 1]$.

**Definition 3.** We say that an NP maximization problem $\Pi$ has a *Polynomial-Time Approximation Scheme (PTAS)* if there is an approximation algorithm that takes as input both an instance $\mathcal{I}$ and a fixed rational parameter $\epsilon > 0$, and outputs a solution which is $(1 - \epsilon)$-approximate in time polynomial in $|\mathcal{I}|$. The class of optimization problems admitting a PTAS algorithm is also denoted by *PTAS*.

**Definition 4.** An NP maximization problem $\Pi_1$ is said to be *AP-reducible* to an NP maximization problem $\Pi_2$ if there are two polynomial-time computable functions $F$ and $G$ and a constant $\alpha$ such that

1. for any instance $\mathcal{I}$ of $\Pi_1$, $F(\mathcal{I})$ is an instance of $\Pi_2$;
2. for any instance $\mathcal{I}$ of $\Pi_1$, and any feasible solution $s'$ of $F(\mathcal{I})$, $G(\mathcal{I}, s')$ is a feasible solution of $\mathcal{I}$;
3. for any instance $\mathcal{I}$ of $\Pi_1$, and any $r \leq 1$, if $s'$ is an $r$-approximate solution of $F(\mathcal{I})$ then $G(\mathcal{I}, s')$ is an $(1 - (1 - r)\alpha - o(1))$-approximate solution of $\mathcal{I}$, where the $o$-notation is with respect to $|\mathcal{I}|$.

An NP maximization problem is *APX-hard* if every problem in APX is AP-reducible to it. It is a well-known fact (see, e.g., [5]) that AP-reductions compose, and that if $\Pi_1$ is AP-reducible to $\Pi_2$ and $\Pi_2$ is in PTAS (resp., APX), then so is $\Pi_1$.

[1] There are other conventions to define the ratio of an approximate solution, e.g., $max\{Val/Opt, Opt/Val\}$ in [5] and $1 - Val/Opt$ in [15]. These conventions are equivalent for maximization problems.

# 3 Max-Sur-CSP($\mathcal{B}$) is in APX

In this section, we first provide a simple algorithm to show that every Max-CSP($\mathcal{B}$) is in APX. Then we adapt this algorithm in order to show that every Max-Sur-CSP($\mathcal{B}$) is also in APX.

**Proposition 5.** Max-CSP($\mathcal{B}$) *is in APX for any finite $\mathcal{B}$.*

*Proof.* We compute a function $h$ from $A$ to $B$ randomly, i.e., for every $a \in A$, choose $h(a)$ uniformly at random from $B$. Every $k_i$-tuple in $R_i^{\mathcal{A}}$ is preserved by $h$ with probability $\frac{|R_i^{\mathcal{B}}|}{|B|^{k_i}}$. Thus we get a randomized $r$-approximation algorithm, where $r = \min_i \frac{|R_i^{\mathcal{B}}|}{|B|^{k_i}}$ is the expected ratio over all random choices. This algorithm can be derandomized via conditional expectations [14]. In fact, it suffices to select at each step the choice with the largest expected number of satisfied constraints. The expected number of satisfied constraints under some partial assignment can be computed in polynomial time. In this way, we obtain a deterministic $r$-approximation algorithm which runs in polynomial time. Thus Max-CSP($\mathcal{B}$) is in APX. $\square$

Since the function $h$ so obtained is not necessarily surjective, the algorithm presented in the proof does not show that Max-Sur-CSP($\mathcal{B}$) is in APX. To resolve this problem, the idea is to fix *some* function values of $h$ at the beginning, and to choose the other function values randomly. We start by the simple case where the signature $\sigma$ consists of only one relation. We have the following lemma:

**Lemma 6.** *For any $0 < r < \frac{|R^{\mathcal{B}}|}{|B|^k}$, there exists a randomized $r$-approximation algorithm for* Max-Sur-CSP($\mathcal{B}$) *which runs in polynomial time.*

*Proof.* Consider the following algorithm, which we call Approx. First, sort the elements in $A$ in increasing order of their *degrees*, which is defined to be the total number of occurrences among all tuples in all relations in $\mathcal{A}$. Then construct an arbitrary bijective function on the set of the first $|B|$ elements in $A$ into the set $B$. Finally, extend this function onto the whole set of $A$ by choosing the other function values uniformly at random. This algorithm runs in polynomial time, and always returns a surjective solution. Let us analyze its approximation performance.

Let $Val$ be the number of preserved $k$-tuples in $R^{\mathcal{A}}$ in our solution, and *Opt* be the maximum possible number of preserved $k$-tuples in $R^{\mathcal{A}}$. Since

the sum of degrees of all elements in $A$ is $k|R^{\mathcal{A}}|$, the sum of the $|B|$ smallest degrees is at most $k|R^{\mathcal{A}}| \cdot \frac{|B|}{|A|}$. So there are at least $(|R^{\mathcal{A}}| - k|R^{\mathcal{A}}| \cdot \frac{|B|}{|A|})$ $k$-tuples in $R^{\mathcal{A}}$ which are not incident with any of the first $|B|$ elements in $A$. Every such $k$-tuple is satisfied with probability $\frac{|R^{\mathcal{B}}|}{|B|^k}$ under uniformly random choices. So we have:

$$\begin{aligned}\frac{\mathbb{E}[Val]}{Opt} &\geq \frac{1}{Opt}\left(|R^{\mathcal{A}}| - k|R^{\mathcal{A}}| \cdot \frac{|B|}{|A|}\right) \cdot \frac{|R^{\mathcal{B}}|}{|B|^k} \\ &\geq \left(1 - k \cdot \frac{|B|}{|A|}\right) \cdot \frac{|R^{\mathcal{B}}|}{|B|^k} \qquad \text{(since } Opt \leq |R^{\mathcal{A}}|\text{)},\end{aligned}$$

where $k$, $|B|$ and $|R^{\mathcal{B}}|$ are constant, only $|A|$ is decided by the input instance. Thus for any $0 < r < \frac{|R^{\mathcal{B}}|}{|B|^k}$, we have a randomized polynomial-time $r$-approximation algorithm: follow the Approx algorithm when $|A| > \frac{k|B|}{1 - r\frac{|B|^k}{|R^{\mathcal{B}}|}}$, and tabulate the solution otherwise. ☐

In general, $\sigma$ consists of $m$ relations $R_1, \ldots, R_m$ with arity $k_1, \ldots, k_m$ respectively. A similar analysis leads to:

$$\begin{aligned}\frac{\mathbb{E}[Val]}{Opt} &\geq \frac{1}{opt}\left(\sum_i |R_i^{\mathcal{A}}| - \left(\sum_i k_i|R_i^{\mathcal{A}}|\right)\frac{|B|}{|A|}\right) \cdot \left(\min_i \frac{|R_i^{\mathcal{B}}|}{|B|^{k_i}}\right) \\ &\geq \left(1 - k_{max} \cdot \frac{|B|}{|A|}\right) \cdot \left(\min_i \frac{|R_i^{\mathcal{B}}|}{|B|^{k_i}}\right) \qquad \text{(since } Opt \leq \sum_i |R_i^{\mathcal{A}}|\text{)}.\end{aligned}$$

Thus we have a randomized polynomial-time $r$-approximation algorithm, for any $0 < r < \min_i \frac{|R_i^{\mathcal{B}}|}{|B|^{k_i}}$.

**Proposition 7.** Max-Sur-CSP($\mathcal{B}$) *is in APX for any finite* $\mathcal{B}$.

*Proof.* We apply again the derandomization via conditional expectation to get a deterministic polynomial-time $r$-approximation algorithm, for any $0 < r < \min_i \frac{|R_i^{\mathcal{B}}|}{|B|^{k_i}}$. ☐

# 4 Inapproximability for Max-Sur-CSP($\mathcal{B}$)

In this section, we show that Max-Sur-CSP($\mathcal{B}$) is *harder* than Max-CSP($\mathcal{B}$) for the same $\mathcal{B}$, in the sense that the approximation ratio of Max-Sur-

CSP($\mathcal{B}$) cannot exceed the approximation ratio of Max-CSP($\mathcal{B}$); and that the APX-hardness of Max-CSP($\mathcal{B}$) implies the APX-hardness of Max-Sur-CSP($\mathcal{B}$).

**Theorem 8.** *Let $r \in (0, 1]$. If* Max-CSP($\mathcal{B}$) *is not polynomial-time $r$-approximable, neither is* Max-Sur-CSP($\mathcal{B}$).

*Proof.* Suppose there is a polynomial-time $r$-approximation algorithm for Max-Sur-CSP($\mathcal{B}$). We will prove that there is also a polynomial-time $r$-approximation algorithm for Max-CSP($\mathcal{B}$), which causes a contradiction.

Given an instance $\mathcal{A}$ of Max-CSP($\mathcal{B}$), construct an instance $\mathcal{A}'$ of Max-Sur-CSP($\mathcal{B}$) as following: extend the underlying domain $A$ to $A'$ by adding $|B|$ new elements, and keep $R_i^{\mathcal{A}'}$ to be the same as $R_i^{\mathcal{A}}$, for every $i \in \{1, .., m\}$. The optimum of Max-Sur-CSP($\mathcal{B}$) with the instance $\mathcal{A}'$ equals that of Max-CSP($\mathcal{B}$) with the instance $\mathcal{A}$. Any $r$-approximate solution of Sur-Max-CSP($\mathcal{B}$) with the instance $\mathcal{A}'$ is also an $r$-approximate solution of Max-CSP($\mathcal{B}$) with the instance $\mathcal{A}$. Since $\mathcal{A}$ is arbitrary, we thus have a polynomial-time $r$-approximation algorithm for Max-CSP($\mathcal{B}$). □

**Proposition 9.** *If* Max-CSP($\mathcal{B}$) *is APX-hard,* Max-Sur-CSP($\mathcal{B}$) *is also APX-hard.*

*Proof.* Given an instance $\mathcal{A}$ of Max-CSP($\mathcal{B}$), construct an instance $\mathcal{A}'$ of Max-Sur-CSP($\mathcal{B}$) as in the above proof. The optimum of Max-Sur-CSP($\mathcal{B}$) with the instance $\mathcal{A}'$ equals that of Max-CSP($\mathcal{B}$) with the instance $\mathcal{A}$. So we have an AP-reduction from Max-CSP($\mathcal{B}$) to Max-Sur-CSP($\mathcal{B}$), where the constant $\alpha$ in the definition of the AP-reduction is 1. Since AP-reductions compose, we then proved the proposition. □

**Corollary 10.** Max-Sur-CSP($C_6$) *is APX-hard and, under the unique games conjecture*[2]*, any polynomial-time approximation ratio of* Max-Sur-CSP($C_6$) *is at most $\alpha_{GW}$ (=0.878. . . ).*

*Proof.* Max-CSP($C_6$) is exactly the same problem as Max-Cut. From the APX-hardness of Max-Cut, we have the APX-hardness of Max-Sur-CSP($C_6$) by Proposition 9. The best approximation ratio of Max-Cut has been proved to be $\alpha_{GW}$, under the unique games conjecture [12]. We then deduce from Theorem 8 that any polynomial-time approximation ratio of Max-Sur-CSP($C_6$) is at most $\alpha_{GW}$. □

[2] A formal description of this conjecture could be found in [11].

# 5 Approximability for Max-Sur-CSP($\mathcal{B}$)

In this section we describe a PTAS for Max-Sur-CSP($\mathcal{B}$), given that Max-CSP($\mathcal{B}$) is in PO. The generalized result is stated in the following theorem.

**Theorem 11.** *Let $r \in (0,1]$. If* Max-CSP($\mathcal{B}$) *is polynomial-time $r$-approximable,* Max-Sur-CSP($\mathcal{B}$) *is polynomial-time $(r-\epsilon)$-approximable, for any $\epsilon > 0$.*

*Proof.* Let Approx1 be a polynomial-time $r$-approximation algorithm for Max-CSP($\mathcal{B}$). Consider first the following randomized algorithm:

Approx2($\mathcal{A}$)
$h \leftarrow$ Approx1($\mathcal{A}$)
**if** $h$ is surjective
  **then return** $h$
  **else** $h^* \leftarrow h$
    **for each** $b \in B$ such that $y$ is not an image of $h^*$
      **do** $T \leftarrow \{x \in A \mid \exists\, y \in A \backslash \{x\}, \text{ s.t. } h^*(x) = h^*(y)\}$
        **if** $T = \emptyset$
          **then return** *No Solution.*
        $x \leftarrow$ an element in $T$ chosen uniformly at random
        $h^*(x) \leftarrow y$
    **return** $h^*$.

Let us analyze the performance of this algorithm. Let $Val$ and $Val^*$ be the number of satisfied constraints in $h$ and $h^*$ respectively. Let $Opt$ and $Opt^*$ be the optimum of Max-CSP($\mathcal{B}$) and Max-Sur-CSP($\mathcal{B}$) respectively. Obviously, $Opt^* \leq Opt$.

When $h$ is surjective, we have $h^* = h$ and $\dfrac{\mathbb{E}[Val^*]}{Opt^*} \geq \dfrac{\mathbb{E}[Val]}{Opt}$.

When $h$ is not surjective, it is more complicated. Let $\delta$ be the number of elements in $B$ which are not images of $h$. For a given tuple $(x_1, \ldots, x_k)$ in $A^k$ and another given tuple $(b_1, \ldots, b_k)$ in $B^k$, define $p$ to be the probability that $h$ maps $(x_1, \ldots, x_k)$ to $(b_1, \ldots, b_k)$ and $p^*$ to be the probability that $h^*$ maps $(x_1, \ldots, x_k)$ to $(b_1, \ldots, b_k)$. Let $T = \{x \in A \mid \exists\, y \in A \backslash \{x\}, \text{ s.t. } h(x) = h(y)\}$ and $t = |T|$. Obviously $t > |A| - |B|$. For every $x \in T$, the algorithm above will not modify $h^*(x)$ if there is no other variables in $A$ with the same function value $h(x)$; otherwise it will modify $h^*(x)$ with probability $\frac{\delta}{t} < \frac{|B|}{|A|-|B|}$. Let $Q$ be the conditional probability that $h^*$ does not map

$(x_1,\dots,x_k)$ to $(b_1,\dots,b_k)$, given that $h$ maps $(x_1,\dots,x_k)$ to $(b_1,\dots,b_k)$. We have:

$$\begin{aligned} Q &\le \textstyle\sum_{i=1}^{k} \mathbb{P}[\text{The algorithm modifies } h^*(x_i)] \\ &\le k \cdot \mathbb{P}[\text{ The algorithm modifies } h^*(x) \text{, for an arbitrary } x \in T] \\ &< k \cdot \frac{|B|}{|A| - |B|}, \end{aligned}$$

which implies that $p^* > \Big(1 - k \cdot \frac{|B|}{|A|-|B|}\Big)p$.

The second line of the inequality above holds because the function value of $h^*$ is modified with the same positive probability at every variable in $T$, and with probability 0 at every variable outside $T$.

Thus we have $\frac{\mathbb{E}[Val^*]}{Opt^*} \ge \frac{\mathbb{E}[Val^*]}{Opt} > \Big(1 - k_{max} \cdot \frac{|B|}{|A|-|B|}\Big)\frac{\mathbb{E}[Val]}{Opt}$, where $k_{max}$ is the maximal arity of all relations.

So whether $h$ is surjective or not, we always have:

$$\frac{\mathbb{E}[Val^*]}{Opt^*} > \Big(1 - k_{max} \cdot \frac{|B|}{|A|-|B|}\Big)\frac{\mathbb{E}[Val]}{Opt}.$$

Hence for any $\epsilon > 0$, we can achieve a randomized $(r-\epsilon)$-approximation algorithm for Max-Sur-CSP($\mathcal{B}$): first, precalculate the optimal assignment for every input relational structure whose domain size is less than $N_0 = \lfloor \frac{r|B|k_{max}}{\epsilon} \rfloor + |B|$. The running time of this part depends only on $\epsilon$, but not on $|\mathcal{A}|$. Then for a given relational structure $\mathcal{A}$, output a precalculated solution if $|A| \le N_0$; and execute Approx2 with the instance $\mathcal{A}$ otherwise.

The algorithm Approx2 can be derandomized by enumerating all possible choices in line 9. This enumerating procedure runs in time polynomial in $|\mathcal{A}|$, since there are polynomially many combinations of choices, and each combination of choices corresponds to a polynomial number of steps in the running time. Thus the derandomized algorithm still runs in time polynomial in $|\mathcal{A}|$. So we get a deterministic polynomial-time $(r-\epsilon)$-approximation algorithm for Max-Sur-CSP($\mathcal{B}$). □

**Proposition 12.** *If* Max-CSP($\mathcal{B}$) *is in PO, then* Max-Sur-CSP($\mathcal{B}$) *is in PTAS and is not APX-hard, unless $P = NP$.*

*Proof.* MAX-CSP($\mathcal{B}$) being in PO implies that it has a polynomial-time approximation algorithm with approximation ratio 1. Theorem 11 then leads to the PTAS-containment. On the other hand, we already know that there are problems in APX but not in PTAS, unless P=NP (see, e.g., [8]). These problems cannot be AP-reduced to MAX-SUR-CSP($\mathcal{B}$), otherwise MAX-SUR-CSP($\mathcal{B}$) could not be in PTAS. So MAX-SUR-CSP($\mathcal{B}$) is not APX-hard, unless P=NP. □

**Corollary 13.** *The Minimum-Asymmetric-Cut problem,* MAX-SUR-CSP($C_4^{\mathrm{ref}}$)*, and the Minimum-Rainbow-Coloring problem are all in PTAS.*

*Proof.* For each of the three problems above, if the function $h$ is not required to be surjective, we can assign 1 to every variable so that all constraints are satisfied. Thus the corresponding MAX-CSP($\mathcal{B}$) problem is in PO. We then conclude by applying Proposition 12. □

# 6 Complexity dichotomy for MAX-SUR-CSP($\mathcal{B}$)

In this section, we first propose a conditional complexity dichotomy for MAX-SUR-CSP($\mathcal{B}$) between PTAS and APX-complete. This condition has already been proved on the Boolean domain and 3-element domains.

**Theorem 14.** *If there is a complexity dichotomy for* MAX-CSP($\mathcal{B}$) *between PO and APX-complete, there is also a complexity dichotomy for* MAX-SUR-CSP($\mathcal{B}$) *between PTAS and APX-complete.*

*Proof.* This result is a combination of Proposition 9 and Proposition 12. □

## 6.1 On the Boolean domain

**Definition 15.** A constraint $R$ is said to be

- *0-valid* if $(0, ..., 0) \in R$.
- *1-valid* if $(1, ..., 1) \in R$.
- *2-monotone* if $R$ is expressible as a DNF-formula either of the form $(x_1 \wedge ... \wedge x_p)$ or $(\overline{y_1} \wedge ... \wedge \overline{y_q})$ or $(x_1 \wedge ... \wedge x_p) \vee (\overline{y_1} \wedge ... \wedge \overline{y_q})$.

A relational structure $\mathcal{B}$ is *0-valid (resp. 1-valid, 2-monotone)* if every constraint in $\mathcal{B}$ is 0-valid (resp. 1-valid, 2-monotone).

Khanna and Sudan have proved that if $\mathcal{B}$ is 0-valid or 1-valid or 2-monotone then MAX-CSP($\mathcal{B}$) is in PO; otherwise it is APX-hard (Theorem 1 in [10]). In fact, when $\mathcal{B}$ is 0-valid (resp. 1-valid), it suffices to assign 0 (resp. 1) to all variables; and when $\mathcal{B}$ is 2-monotone, we can reduce this problem to MIN-CUT, which can be solved efficiently using e.g., the Edmonds-Karp algorithm. Together with Theorem 14 , we have the following result.

**Theorem 16.** *Let $B = \{0, 1\}$. If $\mathcal{B}$ is 0-valid or 1-valid or 2-monotone then* MAX-SUR-CSP($\mathcal{B}$) *is in PTAS; otherwise it is APX-hard.*

Using a similar idea as in the proof of Theorem 1 in [10], we have the following lemma.

**Lemma 17.** *If $\mathcal{B}$ is 2-monotone, then* MAX-SUR-CSP($\mathcal{B}$) *is in PO.*

*Proof.* We reduce the problem of finding the maximum number of satisfiable constraints to the problem of finding the minimum number of violated constraints. This problem, in turn, can be reduced to the $s - t$ MIN-CUT problem in directed graphs. Recall that there are three forms of 2-monotone formulas: (a) $x_1 \wedge ... \wedge x_p$, (b) $\overline{y_1} \wedge ... \wedge \overline{y_q}$, and (c)$(x_1 \wedge ... \wedge x_p) \vee (\overline{y_1} \wedge ... \wedge \overline{y_q})$.

Construct a directed graph $G$ with two special nodes $F$ and $T$ and a different node corresponding to each variable in the input instance. Let $\infty$ denote an integer larger than the total number of constraints. We first select two constraints $C'$ and $C''$, such that $C'$ is of the form (a) or (c), and $C''$ is of the form (b) or (c). Intuitively, $C'$ is to ensure that there exists some element assigned to 1, and $C''$ is to ensure that there exists some element assigned to 0. For a fixed pair $C'$ and $C''$, we proceed as follows for each of the three forms of constraints :

- For a constraint $C$ of the form (a), create a new node $e_C$ and add an edge from each $x_i$ to $e_C$ of cost $\infty$. Add also an edge from $e_C$ to $T$. This edge is of cost $\infty$ if $C$ is selected as $C'$, and of unit cost otherwise.

- For a constraint $C$ of the form (b), create a new node $\overline{e_C}$ and add an edge from $\overline{e_C}$ to each $y_i$ of cost $\infty$. Add also an edge from $F$ to $\overline{e_C}$. This edge is of cost $\infty$ if $C$ is selected as $C''$, and of unit cost otherwise.

- Finally, for a constraint $C$ of the form (c), we create two nodes $e_C$ and $\overline{e_C}$ and connect $e_C$ to each $x_i$ and connect $\overline{e_C}$ to each $y_i$ as described above. If $C$ is selected both as $C'$ and $C''$, then add an edge from $e_C$ to

$T$ of cost $\infty$ and an edge from $F$ to $\overline{e_C}$ of cost $\infty$; otherwise if $C$ is only selected as $C'$, then add an edge from $e_C$ to $T$ of cost $\infty$; otherwise if $C$ is only selected as $C''$, then add an edge from $F$ to $\overline{e_C}$ of cost $\infty$; otherwise add an edge from $e_C$ to $\overline{e_C}$ of unit cost.

From the correspondence between cuts and assignments, by setting variables on the $T$ side of the cut to be 1 and variables on the $F$ side of the cut to be 0, we find that the cost of a minimum cut separating $T$ from $F$, equals the minimum number of constraints that are violated, under the condition that $C'$ and $C''$ are selected to ensure the surjectivity. We only need to go over all possible combinations of $C'$ and $C''$ to achieve the maximum surjective solution. □

We close this section by providing certain MAX-SUR-CSP($\mathcal{B}$) problems which belong to a complexity class different from that of any MAX-CSP($\mathcal{B}$) problem on the Boolean domain or 3-element domains, under the assumption that P≠NP.

**Lemma 18.** *There exist finite relational structures $\mathcal{B}$ such that* MAX-SUR-CSP($\mathcal{B}$) *is NP-hard but in PTAS.*

*Proof.* Let $\mathcal{B}$ be the relational structure on the Boolean domain with the following relation

$$R^{\mathcal{B}} = \{(0,0,0),(0,1,1),(1,0,1),(1,1,0),(0,1,0)\}.$$

This relation is not *weakly positive*, not *weakly negative*, not *affine*, and not *bijunctive*[3], so SUR-CSP($\mathcal{B}$) is NP-hard (which follows from Theorem 4.10 in [4]). Since there is a trivial reduction from SUR-CSP($\mathcal{B}$) to MAX-SUR-CSP($\mathcal{B}$), we have the NP-hardness of MAX-SUR-CSP($\mathcal{B}$). On the other hand, MAX-CSP($\mathcal{B}$) is in PO, since we can assign 0 to every variable to satisfy all constraints. So MAX-SUR-CSP($\mathcal{B}$) is in PTAS by Proposition 12. □

**Remark 19.** *In fact, such $\mathcal{B}$ on the Boolean domain is not unique. Any 0-valid or 1-valid or 2-monotone relational structure which is not weakly positive, not weakly negative, not affine, and not bijunctive satisfies the desired property. There also exist relational structures on larger domains with this property, such as $C_4^{\mathrm{ref}}$ in Example 3.*

[3] Their definitions can be found in [4].

### 6.2 On 3-element domains

Jonsson, Klasson, and Krokhin has proved the complexity dichotomy for MAX-CSP($\mathcal{B}$) between PO and APX-hard on 3-element domains (Theorem 3.1 in [9]). Together with Theorem 14, we have the following result.

**Theorem 20.** *Let $\mathcal{B}$ be any finite relational structure on a 3-element domain.* MAX-SUR-CSP($\mathcal{B}$) *is either in PTAS or APX-hard.*

**Remark 21.** *The detailed statement of this complexity dichotomy needs concepts of cores and supermodularity, see, e.g., [9].*

## 7 Further research

For MAX-CSP($\mathcal{B}$), define its *approximation threshold* $r_0$ to be the supremum of the set of polynomial-time approximation ratios of MAX-CSP($\mathcal{B}$). For any $r > r_0$, MAX-SUR-CSP($\mathcal{B}$) is not polynomial-time $r$-approximable (by Proposition 8); and for any $r < r_0$, MAX-SUR-CSP($\mathcal{B}$) is polynomial-time $r$-approximable (by Proposition 11). What is the complexity of MAX-SUR-CSP($\mathcal{B}$) when the desired approximation ratio is $r_0$?

**Open Problem 1.** Let $r_0$ be the approximation threshold of MAX-CSP($\mathcal{B}$). Is MAX-SUR-CSP($\mathcal{B}$) polynomial-time $r_0$-approximable?

Consider a special case of the above problem, where $r_0$ is 1, which means that MAX-CSP($\mathcal{B}$) is in PO.

**Open Problem 2.** Given that MAX-CSP($\mathcal{B}$) is in PO. Is MAX-SUR-CSP($\mathcal{B}$) in PO?

In fact, we failed to give an answer, even for some concrete finite structures $\mathcal{B}$ on small domains:

**Open Problem 3.** Are the Minimum-Asymmetric-Cut problem and the Minimum-Rainbow-Coloring problem in PO?

The difficulty of the above problems lies in the classification of the MAX-SUR-CSP($\mathcal{B}$) problems which are in PTAS.

**Open Problem 4.** If P≠NP, is there a complexity trichotomy for MAX-SUR-CSP($\mathcal{B}$) among PO, PTAS but NP-hard, and APX-hard, for any finite structure $\mathcal{B}$ on the Boolean domain or 3-element domains?

# References


[1] M. Bodirsky. Constraint satisfaction with infinite domains. PhD thesis, Humboldt-Universitat zu Berlin, 2004.

[2] M. Bodirsky, J. Kára, and B. Martin. The complexity of surjective homomorphism problems – a survey. Preprint, arXiv:1104.5257, 2011.

[3] A. A. Bulatov. A dichotomy theorem for constraint satisfaction problems on a 3-element set. *Journal of the ACM*, 53(1):66–120, 2006.

[4] N. Creignou and J. J. Hébrard. On generating all solutions of generalized satisfiability problems. *Informatique Thèorique et Applications*, 31(6):499–511, 1997.

[5] N. Creignou, S. Khanna, and M. Sudan. *Complexity Classifications of Boolean Constraint Satisfaction Problems*. SIAM Monographs on Discrete Mathematics and Applications, 2001.

[6] T. Feder, P. Hell, and J. Huang. List homomorphisms and circular arc graphs. *Combinatorica*, 19(4):487–505, 1999.

[7] T. Feder and M. Vardi. Monotone monadic SNP and constraint satisfaction. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, pages 612–622, 1993.

[8] J. Håstad. Some optimal inapproximability results. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, pages 1–10, 1997.

[9] P. Jonsson, M. Klasson, and A. Krokhin. The approximability of three-valued MAX CSP. *SIAM Journal on Computing*, 35(6):1329–1349, 2006.

[10] S. Khanna and M. Sudan. The optimization complexity of constraint satisfaction problems. Technical report, Stanford University, Stanford, CA, USA, 1996.

[11] S. Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 767–775, 2002.

[12] S. Khot, G. Kindler, E. Mossel, and R. O'Donnell. Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 146–154, 2004.

[13] B. Martin and D. Paulusma. The computational complexity of disconnected cut and 2K2-partition. *Principles and Practice of Constraint Programming*, pages 561–575, 2011.

[14] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[15] C. H. Papadimitriou. *Computational Complexity*. Addison Wesley, 1994.

[16] T. J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing*, pages 216–226, 1978.